OMCP: Model Context Protocol Servers for the OMOP Common Data Model

Shihao Shenzhang[1], Niko Möller-Grell[1], Zhangshu Joshua Jiang[6], Richard Dobson[1,2,3,4,5], Vishnu V Chandrabalan[6,7]

1. **Department of Biostatistics and Health Informatics, Institute of Psychiatry, Psychology and Neuroscience (IoPPN), King's College London, London, UK**
2. **Institute for Health Informatics, University College London, London, UK**
3. **NIHR Biomedical Research Centre, University College London Hospitals National Health Service Foundation Trust, London, UK**
4. **Health Data Research UK London, University College London, London, UK**
5. **NIHR Biomedical Research Centre, South London and Maudsley National Health Service Foundation Trust and King's College London, London, UK**
6. **Lancashire and South Cumbria Secure Data Environment**
7. **Lancaster University**

Background

The Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) has emerged as a global standard for generating real-world evidence (RWE). It achieves this by harmonising routinely collected electronic health record (EHR) data into a relational structure and mapping clinical terms to standardised ontologies.[1] The OMOP standard and its suite of open-source analytical tools are maintained by the Observational Health Data Sciences and Informatics (OHDSI) collaborative. In the UK, the NHS England Secure Data Environment network has adopted the OMOP CDM, and Oxford University—together with Health Data Research UK (HDRUK)—has established the country's first RWE network, known as HERON.

Recent advances in large language models (LLMs), such as OpenAI's GPT-4o and Anthropic's Claude Sonnet 3.7, offer new opportunities for healthcare research, particularly in data analysis and clinical insight generation.[2] These models can translate natural language prompts into SQL queries or analytics code across multiple programming languages. Lightweight, quantised LLMs—available through platforms such as Ollama—now enable local deployment with reduced computational overhead, making them suitable for academic and healthcare institutions. Local deployment also mitigates governance and privacy concerns by ensuring that sensitive data remains within institutional boundaries.

In November 2024, Anthropic introduced the Model Context Protocol (MCP), an open standard that enables LLMs to interact with contextual "tools" during user sessions.[3] Since then, several open-source MCP servers have emerged to facilitate LLM access to real-world data.[4] The protocol has since evolved to support agentic workflows through integrations like Google's A2A protocol, allowing LLMs to plan and execute multi-step tasks across tools.[5] These agentic capabilities go beyond single-step queries, enabling models to reason over intermediate results, orchestrate toolchains, and adapt to new information in real-time, a feature especially valuable for complex domains such as healthcare.

Despite these advances, generating meaningful insights from OMOP databases still requires expertise in SQL and R. While some LLM-based tools have been developed to support software engineering (e.g., Claude Code, Cursor) or scientific exploration (e.g., Google's AI co-scientist), there are currently no privacy-preserving solutions tailored to accelerate RWE generation from OMOP datasets. This represents a key gap for the development of LLM-integrated systems in healthcare analytics.

We introduce OMCP[6], a suite of MCP servers that integrates the OMOP CDM with MCP to extend LLM capabilities beyond basic chat interactions. By enabling real-time natural language-to-database operations specific to OMOP (e.g., on-demand analytics, dynamic querying), OMCP empowers clinicians and researchers to generate insights rapidly without SQL expertise while preserving the flexibility for advanced analyses. In this abstract, we present OMCP-SQL, the first in a series of OMOP-specific MCP servers, comprising the OMCP Semantic Parser, OMCP Python, OMCP Data validator and an R Sandbox—all interconnected through the OMCP-A2A framework to enable collaborative agentic workflows across tools.

Methods

OMCP-SQL was developed using multiple open-source Python libraries and tested against a synthetic OMOP database derived from the publicly available Synthea dataset. While existing SQL MCP servers, such as Postgres MCP servers, offer generic database access to LLMs, OMCP-SQL was built with multiple additional capabilities as described below.

As illustrated in Figure 1, SQLGlot was used for syntactic and semantic validation of LLM-generated queries against user-configurable criteria.[6] This allowed for both fine-grained control over what queries were sent to the database engine for execution and the raising of applicable "LLM-friendly" exceptions when the SQL generated by LLMs did not pass validation. Syntactic validation prevents poor queries from being submitted to the data warehouse, potentially reducing the execution cost, especially for cloud-hosted systems. Semantic validation enables the filtering of queries to avoid destructive actions (e.g., INSERT, TRUNCATE) from being sent to the data warehouse, thereby reducing the risk of misconfigured access controls. Semantic validation also allows the user to configure which tables and columns the LLM may access.

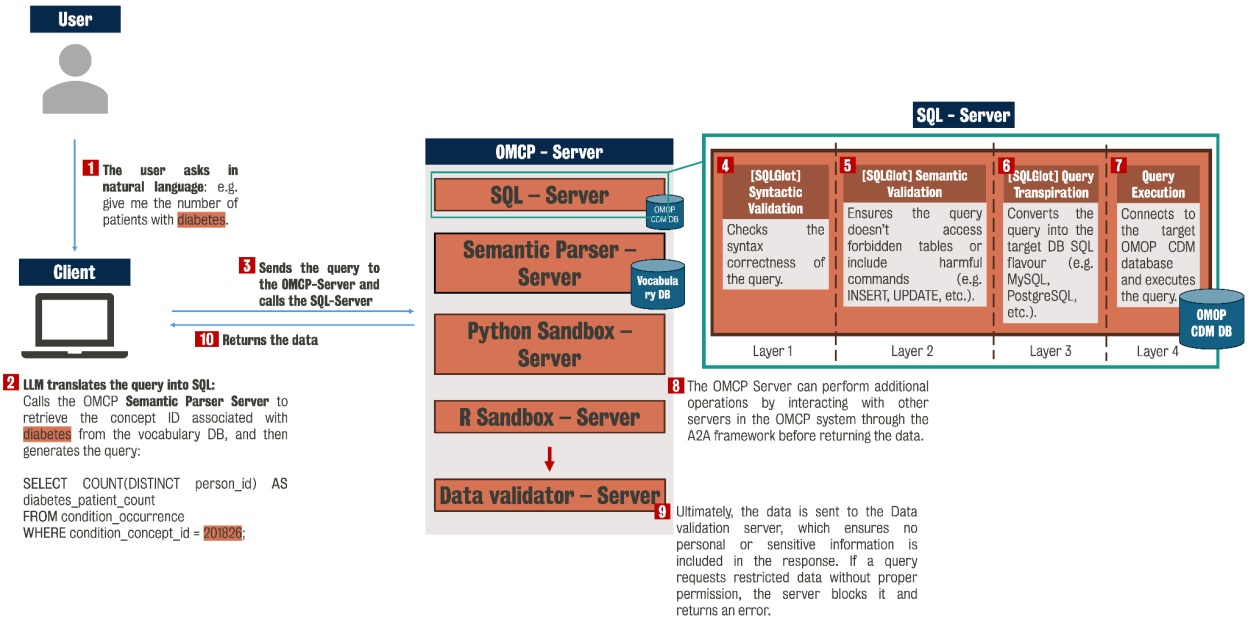

Figure 1: OMCP server information flow diagram, with a zoom-in on the SQL-Tool. Multiple layers of security checks are performed using SQLGlot and Ibis before the query is executed against the database.

To support diverse OMOP deployments, SQLGlot is combined with the Ibis framework for SQL

transpilation and dataframe semantics, enabling compatibility with multiple database backends.[7] [8]

Finally, a Data Validation Tool ensures privacy and access control at the response level. It verifies that OMCP outputs do not include sensitive or personally identifiable information. Where applicable, it anonymises and generalises results; otherwise, it returns an error to prevent disclosure.

Results

OMCP-SQL was evaluated using both local large language models (LLMs) and Anthropic's Claude Sonnet 3.7. Local testing was performed with Qwen2.5-Coder:14B (Alibaba) and Cogito:14B, both served through Ollama v0.6.8, with Oterm[9] and LibreChat[10] acting as user-facing interfaces. For testing Anthropic models, we used Claude Desktop.

A subset of 15 query descriptions from the ACHILLES suite (Automated Characterisation of Health Information at Large-scale Longitudinal Evidence Systems) was used to generate natural language prompts aimed at eliciting SQL queries for clinically relevant analyses. To evaluate system robustness, adversarial prompts were also crafted to induce destructive operations (e.g., INSERT, DELETE) or access restricted tables (e.g., METADATA).

System performance was evaluated based on the LLMs' ability to translate prompts into safe and valid SQL queries. Table 1 presents the results of this evaluation. Among the models tested, Claude Sonnet achieved the highest success rate at 93.3%, with an average of 4.73 attempts per query. The number of attempts reflects how often the system queried the database for context before generating a final, valid output.

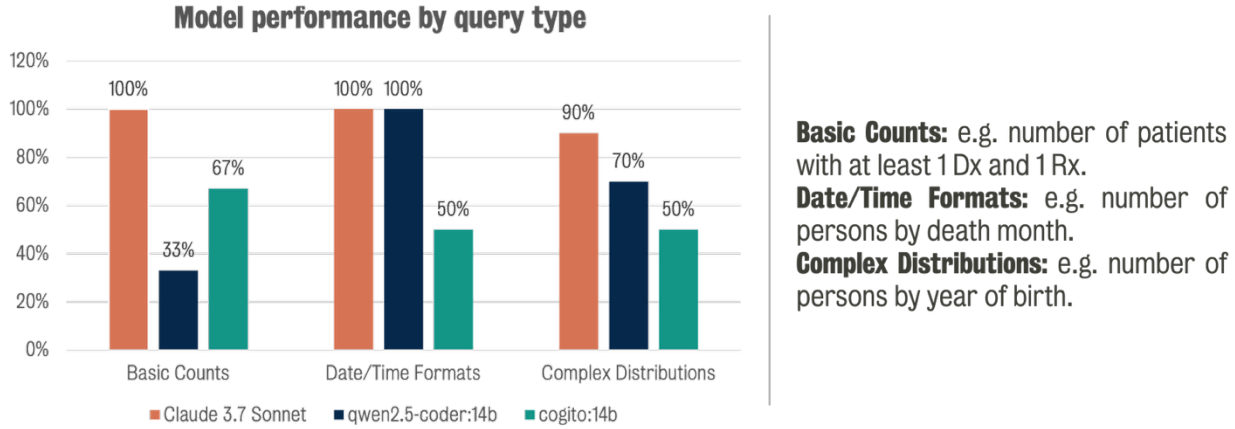

Table 1: Model comparison with different types of prompts from ACHILLES provided to the model to evaluate its performance.

Conclusion

Our work on OMCP demonstrates the potential of large language models to bridge the accessibility gap between clinicians, researchers, and the OMOP Common Data Model. By combining the semantic SQL parsing capabilities of SQLGlot with the contextual reasoning of LLMs, we developed a system that enables natural language interaction with OMOP. Rather than building a domain-specific model, OMCP

uses general-purpose LLMs via the standardised MCP protocol. This approach allows for the OHDSI community to integrate agentic frameworks, such as OMCP-A2A, with locally deployed LLMs, thereby reducing the risk of data leakage to commercial API endpoints.

References

1. Ohdsi. The book of OHDSI Observational Health Data Sciences and Informatics. San Bernardino, Ca Ohdsi; 2019.
2. Yu P, Xu H, Hu X, Deng C. Leveraging Generative AI and Large Language Models: A Comprehensive Roadmap for Healthcare Integration. Healthcare [Internet]. 2023 Jan 1;11(20):2776. Available from: https://www.mdpi.com/2227-9032/11/20/2776
3. Introduction - Model Context Protocol [Internet]. Modelcontextprotocol.io. Model Context Protocol; 2025. Available from: https://modelcontextprotocol.io/introduction
4. MCP Server Directory: 4230+ updated daily | PulseMCP [Internet]. PulseMCP. 2025 [cited 2025 May 9]. Available from: https://www.pulsemcp.com/servers
5. A2A Project. A2A: Agent-to-Agent Protocol [Internet]. 2024 [cited 2025 Jun 29]. Available from: https://a2aproject.github.io/A2A/latest/
6. vvcb. OMOP MCP Server [Internet]. Github.io. 2025 [cited 2025 May 9]. Available from: https://fastomop.github.io/omcp/
7. sqlglot API documentation [Internet]. Sqlglot.com. 2023 [cited 2025 May 9]. Available from: https://sqlglot.com/sqlglot.html
8. Ibis [Internet]. Ibis. 2025 [cited 2025 May 9]. Available from: https://ibis-project.org/
9. oterm - oterm [Internet]. Github.io. 2025 [cited 2025 May 9]. Available from: https://ggozad.github.io/oterm/
10. LibreChat [Internet]. Librechat.ai. 2025. Available from: https://www.librechat.ai/